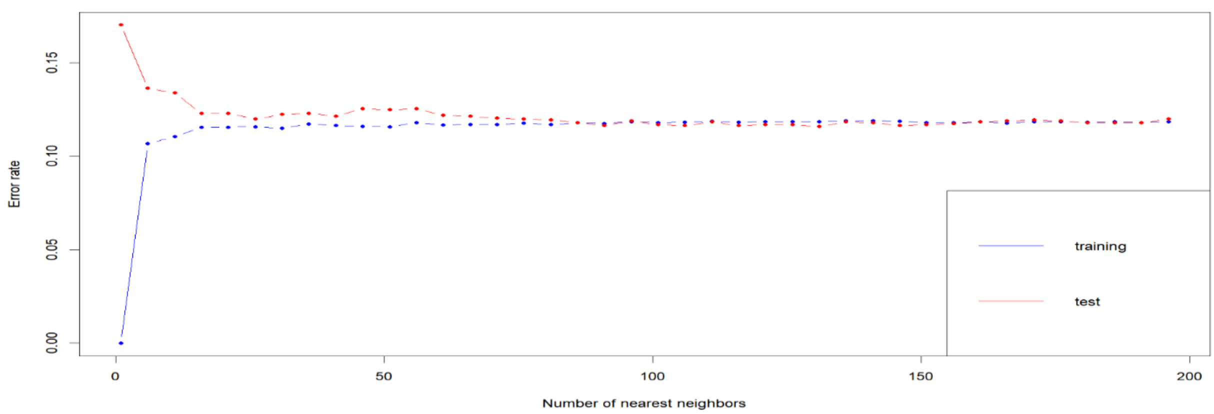
STAT 4360 (Introduction to Statistical Learning, Fall 2022)
Mini Project 1
Name: Haeun Kim

Section 1

1)

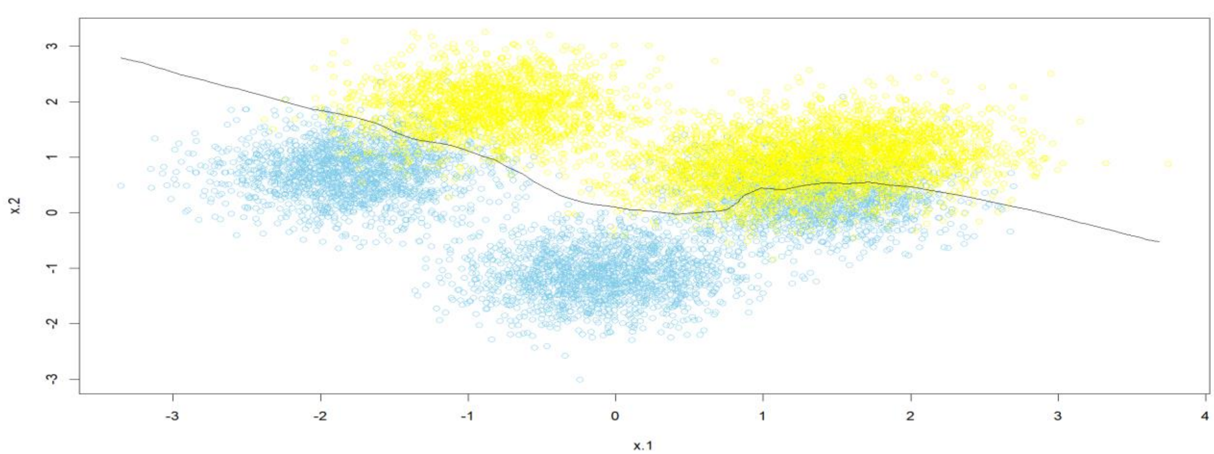
(a) First of all, I did figure out the sequence by using sequence function from 1 to 200 by 5 because K = 1, 6, 11, …., 200. By using knn function, I could fit the KNN with those K numbers by setting the seed as 100.

(b) What I can observe is the error rate of test (over 0.15) is much higher than error rate of training (as 0) at first. However, as the number of nearest neighbors (K) increase, the error rate of training tends to increase, while the error rate of test tends to decrease. Therefore, they seem to stabilize (like a line) in some error rate after the number of nearest neighbors becomes around 90. When the number of nearest neighbors is around 100 (-20 to +20), the error rate of training becomes a little higher than the error rate of test. However, the test error rate reaches the minimum error rate at optimal K, and it increases afterward. Furthermore, in some number of nearest neighbors around, we could notice test error rate becomes a little higher than the error rate of training. This observation is consistent with what I've expected from the class. The training error rate increases and the test error rate decreases, but the test error rate reaches the lowest error rate and increases again. So, as K increases, the flexibility increases because the training error rate decreases, and the test error rate shows a slight U-shape



(c) Optimal value of K what I found is 131 by using the minimum of test error rate. The training error rate is 0.1185, and the test error rate is 0.1160.

(d)



(e)

2)

(a) $MSE\{\hat{f}(x_0)\} = [Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\}$

LHS)

$$
\begin{aligned}
MSE\{\hat{f}(x_0)\} &= E\{(\hat{f}(x_0) - f(x_0)\}^2 \\
&= E\{\hat{f}(x_0)^2 - \hat{f}(x_0)f(x_0) - \hat{f}(x_0)f(x_0) + f(x_0)^2\} \\
&= E\{\hat{f}(x_0)^2\} - f(x_0)E\{\hat{f}(x_0)\} - f(x_0)E\{\hat{f}(x_0)\} + f(x_0)^2 \\
&= E\{\hat{f}(x_0)^2\} - 2f(x_0)E\{\hat{f}(x_0)\} + f(x_0)^2
\end{aligned}
$$

RHS)

$$
\begin{aligned}
[Bias\{\hat{f}(x_0)\}]^2 &= [E\left(\hat{f}(x_0)\right) - f(x_0)]^2 \\
&= [E\left(\hat{f}(x_0)\right)]^2 - 2f(x_0)E\{\hat{f}(x_0)\} + f(x_0)^2
\end{aligned}
$$

$$
\begin{aligned}
var\{\hat{f}(x_0)\} &= E\{\hat{f}(x_0) - E(\hat{f}(x_0))^2 \\
&= E\{\left(\hat{f}(x_0)\right)^2 - 2\hat{f}(x_0)E\{\hat{f}(x_0)\} + \left\{E\left(\hat{f}(x_0)\right)\right\}^2\} \\
&= E\left\{\left(\hat{f}(x_0)\right)^2\right\} - 2E\{\hat{f}(x_0)\}E\{\hat{f}(x_0)\} + E[E\{\hat{f}(x_0)\}]^2 \\
&= E\left\{\left(\hat{f}(x_0)\right)^2\right\} - 2[E\{\hat{f}(x_0)\}]^2 + [E\{\hat{f}(x_0)\}]^2 \\
&= E\left\{\left(\hat{f}(x_0)\right)^2\right\} - [E\{\hat{f}(x_0)\}]^2
\end{aligned}
$$

So, $[Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\} = E\left\{\left(\hat{f}(x_0)\right)^2\right\} - 2f(x_0)E\{\hat{f}(x_0)\} + f(x_0)^2$

Therefore, the Left-Hand Side (MSE) and the Right-Hand Side ($Bias^2 + var$) are same.

Therefore, $MSE\{\hat{f}(x_0)\} = Bias\{\hat{f}(x_0)\}^2 + var\{\hat{f}(x_0)\}$ is established.

(b) $E\{\hat{Y}_o - Y_0\}^2 = [Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\} + \sigma^2$

LHS)

$E\{\hat{Y}_o - Y_0\}^2 = E\{\hat{f}(x_0) - Y_0\}^2 = E\left\{\left(\hat{f}(x_0) - f(x_0)\right) + (f(x_0) - Y_0)\right\}^2$

$= E\left\{\left(\hat{f}(x_0) - f(x_0)\right) + (-\epsilon_0)\right\}^2$ → $f(x_0) - f(x_0) - \epsilon_0$ because $Y_0 = f(x_0) + \epsilon_0$

$= E\{(\hat{f}(x_0) - f(x_0))^2 - 2\,\epsilon_0\left(\hat{f}(x_0) - f(x_0)\right) + \epsilon_0{}^2\}$

$= E\{(\hat{f}(x_0) - f(x_0))^2 - 2E\left\{\epsilon_0\left(\hat{f}(x_0) - f(x_0)\right)\right\} + E(\epsilon_0{}^2)$

$= E\{(\hat{f}(x_0) - f(x_0)\}^2 - 2E(\epsilon_0)E\left(\hat{f}(x_0) - f(x_0)\right) + E(\epsilon_0{}^2)$ ---→ $E(\epsilon_0) = 0$

$= E\{(\hat{f}(x_0) - f(x_0))^2 + \sigma^2$ ---→ $E(\epsilon_0{}^2) = \sigma^2$

$= MSE\{\hat{f}(x_0)\} + \sigma^2$ ---→ $E\{(\hat{f}(x_0) - f(x_0))^2 = MSE\{\hat{f}(x_0)\}$

$= [Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\} + \sigma^2$ ---→ $MSE\{\hat{f}(x_0)\} = [Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\}$

Therefore, the Left-Hand Side finally reached to Right-Hand Side

So, $E\{\hat{Y}_o - Y_0\}^2 = [Bias\{\hat{f}(x_0)\}]^2 + var\{\hat{f}(x_0)\} + \sigma^2$ is established.

(c) When the model flexibility increases, the square bias decreases and the variance increases. The square bias and the variance chane at different rate. There is usually a "U" shape for the test MSE, due to the change at different rate.

Code

```
library(class)
library(ISLR2)

setwd("C:/Users/haeun/OneDrive/문서/STAT33550")

# Read the test data and training data respectively
test <- read.csv("1-test_data.csv")
train <- read.csv("1-training_data.csv")

# Assign Y for train and test individually
# from the data set train and test
train_Y <- train$y
test_Y <- test$y

# Delete the y column for two data
train_X <- train[, -3]
test_X <- test[, -3]

#Question 1-(a)
# K number should be positive from 1 to 200 by 5
k_series <- seq(1, 200, by = 5)
# Figuring out the number of Ks
k_num <- length(k_series)

# Initialize the error rate for train and test
err_rate_train <- numeric(length = k_num)
err_rate_test <- numeric(length = k_num)
# To name the index by K series
names(err_rate_train) <- k_series
names(err_rate_test) <- k_series

# To figure out the KNN for all the series numbers
# set the set.seed as 1
# Figuring out the error rate for train and test repectively
# By using conditional probability
for(i in seq(along = k_series)){
  set.seed(100)
  mod_train <- knn(train_X, train_X, train_Y, k = k_series[i])
  set.seed(100)
  mod_test <- knn(train_X, test_X, train_Y, k = k_series[i])
  err_rate_train[i] <- 1 - sum(mod_train == train_Y) / length(train_Y)
  err_rate_test[i] <- 1 - sum(mod_test == test_Y) / length(test_Y)
}

# Question 1-(b)
# Plotting the train error rate and test error rate by number of
nearest neighbors
```

```
plot(k_series, err_rate_train,
     xlab = "Number of nearest neighbors", ylab = "Error rate",
     type = "b", ylim = range(c(err_rate_train, err_rate_test)), col =
"blue", pch = 20
)
lines(k_series, err_rate_test, type = "b", col = "red", pch = 20)
legend("bottomright", lty = 1, col = c("blue", "red"), legend =
c("training", "test"))

# Question 1-(c)
# The result is the data frame for the the K number, training error
rate, and test error rate
result <- data.frame(k_series, err_rate_train, err_rate_test)

# I could find the minimum optimal number by figuring out the minimum
test error rate
min_optimal <- which.min(result$err_rate_test)
# Find the index for the minimum of test error rate
optimal_K <- result$k_series[min_optimal]
# Figuring out the test error rate associated with optimal K
err_rate_test_optK <- result$err_rate_test[min_optimal]
# Figuring out the train error rate associated with optimal K
err_rate_train_optK <- result$err_rate_train[min_optimal]

# Question 1-(d)
#  a plot of the training data with Decision boundary for Optimal K
# the size should be 50 X 50, and including all the value of train
data
x1_grid <- seq(f = min(train_X[, 1]), t = max(train_X[, 1]), l = 50)
x2_grid <- seq(f = min(train_X[, 2]), t = max(train_X[, 2]), l = 50)
grid <- expand.grid(x1_grid, x2_grid)

# Set the seed as 100
# Figuring out the KNN for the optimal K
set.seed(100)
mod_opt <- knn(train_X, grid, train_Y, k = optimal_K, prob = T)

# Voting fraction for winning class and Y = "yes"
prob <- attr(mod_opt, "prob")
prob <- ifelse(mod.opt == "yes", prob, 1 - prob)

prob <- matrix(prob, n.grid, n.grid)

# Scatter plot for the training data set
plot(train_X, col = ifelse(train_Y == "yes", "skyblue", "yellow"))

# Contouring to make the boundary
contour( x1_grid, x2_grid, prob,
        levels = 0.5, labels = "", xlab = "X1", ylab = "X2",
        main = "The Decision Boundary Plot", add = T)
```